AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application.

1. (Currently Amended) A time-stationary processor arranged for execution of a program, the processor comprising:

a plurality of execution units, comprising at least a first execution unit and a second execution unit;

a register file accessible by the execution units;

a communication network for coupling the execution units and the register file;

a controller arranged for controlling the processor based on control information derived from the program,

characterized in that at least the first execution unit and the second execution unit are each arranged to produce, according to a programmed criterion specified within said program, a second identifier on validity of an output result for corresponding output ports of the first and second execution units, the processor being arranged to dynamically control writing of result data corresponding to an operation into the register file, based at least on the second identifier and without performing a jump operation.

2. (Currently amended) A processor according to claim 1, characterized in that the control information further comprises a first identifier on the validity of the operation, the first identifier indicating that the operation is invalid in a case where the operation is a NOP operation, and wherein the processor is arranged to dynamically control writing of result data corresponding to the operation into the register file, based on both the first identifier and the second identifier.

3. (Original) A processor according to claim 2, characterized in that the first identifier is delayed according to the pipeline of the corresponding execution unit arranged for executing the operation.

4. (Canceled).

5. (Previously presented) A processor according to claim 3, characterized in that the processor is further arranged to dynamically control writing of result data corresponding to the operation into the register file, based on the first identifier, the second identifier and an input datum.

6. (Original) A processor according to claim 1, characterized in that the register file is a distributed register file.

7. (Original) A processor according to claim 1, characterized in that the communication network is a partially connected communication network.

8. (Currently Amended) A method for controlling a time-stationary processor arranged for execution of a program, wherein the processor comprises:

a plurality of execution units, comprising at least a first execution unit and a second execution unit;

a register file accessible by the execution units;

a communication network for coupling the execution units and the register file;

a controller arranged for controlling the processor based on control information derived from the program,

characterized in that the method for controlling the time-stationary processor comprises the steps of:

producing, by both the first execution unit and the second execution unit, according to a programmable criterion specified within said program, a second identifier on the validity of an output result for a corresponding output port of the first execution unit and the second execution unit, and

dynamically controlling the writing of result data corresponding to an operation into the register file, based at least on the second identifier and without performing a jump operation.

9. (Currently amended) The method of claim 8, further comprising the step of providing a first identifier on the validity of the operation, the first identifier indicating that the operation is invalid in a case where the operation is a NOP operation, and carrying out the step of dynamically controlling the writing of result data corresponding to the operation into the register file, based on both the first identifier and the second identifier.

10. (Previously presented) The method of claim 9, further comprising the step of delaying the first identifier according to the pipeline of the corresponding execution unit arranged for executing the operation.

11. (Previously presented) The method of claim 10, further comprising dynamically controlling writing of result data corresponding to the operation into the register file, based on the first identifier, the second identifier and an input datum.

12. (Previously presented) The method of claim 8, further comprising providing the register file as a distributed register file.

13. (Previously presented) The method of claim 8, further comprising providing the communication network as a partially connected communication network.